

```

# This program adds two values stored in memory.
# This is not a particularly general purpose program, but
# it illustrates how some of the MIPS instructions operate.
# The system call codes are described in Appendix A, p A-43.

        .data    # The following entries go in the data segment
eol:
        .asciiz  "\n"
strA:
        .asciiz  " plus "
strB:
        .asciiz  " equals "
val:
        .word    2,3

        .text    # The following entries go in the text (program code) segment
main:
        la      $a0,eol          # load address of the string to print
        li      $v0,4           # load print_string code
        syscall

        lw      $a0,val         # put the first value in $a0 for printing
        li      $v0,1           # load print_integer code
        syscall

        la      $a0,strA        # load address of the string to print
        li      $v0,4           # load print_string code
        syscall

        lw      $a0,val+4       # put the second value in $a0 for printing
        li      $v0,1           # load print_integer code
        syscall

        la      $a0,strB        # load address of the string to print
        li      $v0,4           # load print_string code
        syscall

        lw      $t0,val         # load first value to add
        lw      $t1,val+4       # load second value to add
        add     $a0,$t0,$t1     # add and put result in $a0
        li      $v0,1           # load print_integer code
        syscall

        la      $a0,eol         # load address of the string to print
        li      $v0,4           # load print_string code
        syscall

        jr      $ra            # return

```

Assembled version of addvals.s

```

[0x00400000] 0x8fa40000 lw $4, 0($29) ; 175: lw $a0 0($sp) # argc
[0x00400004] 0x27a50004 addiu $5, $29, 4 ; 176: addiu $a1 $sp 4 # argv
[0x00400008] 0x24a60004 addiu $6, $5, 4 ; 177: addiu $a2 $a1 4 # envp
[0x0040000c] 0x00041080 sll $2, $4, 2 ; 178: sll $v0 $a0 2
[0x00400010] 0x00c23021 addu $6, $6, $2 ; 179: addu $a2 $a2 $v0
[0x00400014] 0x0c100009 jal 0x00400024 [main] ; 180: jal main
[0x00400018] 0x00000000 nop ; 181: nop
[0x0040001c] 0x3402000a ori $2, $0, 10 ; 183: li $v0 10
[0x00400020] 0x0000000c syscall ; 184: syscall # syscall 10 (exit)
[0x00400024] 0x3c041001 lui $4, 4097 [eol] ; 18: la $a0,eol # load address of the string to print
[0x00400028] 0x34020004 ori $2, $0, 4 ; 19: li $v0,4 # load print_string code
[0x0040002c] 0x0000000c syscall ; 20: syscall # ask the system to print the string
[0x00400030] 0x3c011001 lui $1, 4097 ; 22: lw $a0,val # put the first value in $a0 for printing
[0x00400034] 0x8c240014 lw $4, 20($1)
[0x00400038] 0x34020001 ori $2, $0, 1 ; 23: li $v0,1 # load print_integer code
[0x0040003c] 0x0000000c syscall ; 24: syscall # ask the system to print the integer
[0x00400040] 0x3c011001 lui $1, 4097 [strA] ; 26: la $a0,strA # load address of the string to print
[0x00400044] 0x34240002 ori $4, $1, 2 [strA]
[0x00400048] 0x34020004 ori $2, $0, 4 ; 27: li $v0,4 # load print_string code
[0x0040004c] 0x0000000c syscall ; 28: syscall # ask the system to print the string
[0x00400050] 0x3c011001 lui $1, 4097 ; 30: lw $a0,val+4 # put the second value in $a0 for printing
[0x00400054] 0x8c240018 lw $4, 24($1)
[0x00400058] 0x34020001 ori $2, $0, 1 ; 31: li $v0,1 # load print_integer code
[0x0040005c] 0x0000000c syscall ; 32: syscall # ask the system to print the integer
[0x00400060] 0x3c011001 lui $1, 4097 [strB] ; 34: la $a0,strB # load address of the string to print
[0x00400064] 0x34240009 ori $4, $1, 9 [strB]
[0x00400068] 0x34020004 ori $2, $0, 4 ; 35: li $v0,4 # load print_string code
[0x0040006c] 0x0000000c syscall ; 36: syscall # ask the system to print the string
[0x00400070] 0x3c011001 lui $1, 4097 ; 38: lw $t0,val # load first value to add
[0x00400074] 0x8c280014 lw $8, 20($1)
[0x00400078] 0x3c011001 lui $1, 4097 ; 39: lw $t1,val+4 # load second value to add
[0x0040007c] 0x8c290018 lw $9, 24($1)
[0x00400080] 0x01092020 add $4, $8, $9 ; 40: add $a0,$t0,$t1 # add and put result in $a0
[0x00400084] 0x34020001 ori $2, $0, 1 ; 41: li $v0,1 # load print_integer code
[0x00400088] 0x0000000c syscall ; 42: syscall # ask the system to print the string
[0x0040008c] 0x3c041001 lui $4, 4097 [eol] ; 44: la $a0,eol # load address of the string to print
[0x00400090] 0x34020004 ori $2, $0, 4 ; 45: li $v0,4 # load print_string code
[0x00400094] 0x0000000c syscall ; 46: syscall # ask the system to print the string
[0x00400098] 0x03e00008 jr $31 ; 48: jr $ra # return

```

Registers after running addvals.s

PC = 00000000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000  
Status = 3000ff10 HI = 00000000 LO = 00000000

General Registers

R0 (r0) = 00000000 R8 (t0) = 00000002 R16 (s0) = 00000000 R24 (t8) = 00000000  
R1 (at) = 10010000 R9 (t1) = 00000003 R17 (s1) = 00000000 R25 (t9) = 00000000  
R2 (v0) = 0000000a R10 (t2) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000  
R3 (v1) = 00000000 R11 (t3) = 00000000 R19 (s3) = 00000000 R27 (k1) = 00000000  
R4 (a0) = 10010000 R12 (t4) = 00000000 R20 (s4) = 00000000 R28 (gp) = 10008000  
R5 (a1) = 7ffffef44 R13 (t5) = 00000000 R21 (s5) = 00000000 R29 (sp) = 7ffffef40  
R6 (a2) = 7ffffef48 R14 (t6) = 00000000 R22 (s6) = 00000000 R30 (s8) = 00000000  
R7 (a3) = 00000000 R15 (t7) = 00000000 R23 (s7) = 00000000 R31 (ra) = 00400018

Data segment

DATA  
[0x10000000]...[0x10010000] 0x00000000  
[0x10010000] 0x7020000a 0x2073756c 0x71652000 0x736c6175  
[0x10010010] 0x00000020 0x00000002 0x00000003 0x00000000  
[0x10010020]...[0x10040000] 0x00000000